LXService: Web Services of Language Technology for Portuguese

António Branco, Francisco Costa, Pedro Martins, Filipe Nunes, João Silva, Sara Silveira
University of Lisbon

Dep. Informática, Faculdade de Ciências de Lisboa, Campo Grande, 1749-016 Lisboa, Portugal
{Antonio.Branco, fcosta, pmartins, fnunes, jsilva, sara.silveira}@di.fc.ul.pt
http://nlx.di.fc.ul.pt

Abstract

In the present paper we report on the development of a cluster of web services of language technology for Portuguese that we named as LXService. These web services permit the direct interaction of client applications with language processing tools via the Internet.

This way of making available language technology was motivated by the need of its integration in an eLearning environment. In particular, it was motivated by the development of new multilingual functionalities that were aimed at extending a Learning Management System and that needed to resort to the outcome of some of those tools in a distributed and remote fashion.

This specific usage situation happens however to be representative of a typical and recurrent set up in the utilization of language processing tools in different settings and projects. Therefore, the approach reported here offers not only a solution for this specific problem, which immediately motivated it, but contributes also some first steps for what we see as an important paradigm shift in terms of the way language technology can be distributed and find a better way to unleash its full potential and impact.

1. Introduction

In this paper we present the achievements obtained in the development of a cluster of web services of language technology for Portuguese.

The development of these web services started in the scope of the LT4eL-Language Technology for e-Learning project. They are supported by a range of language technology tools that have been developed in the past at the University of Lisbon in the scope of a number of previous projects.

The LT4eL project aims at using multilingual language technology tools and semantic web techniques for supporting e-Learning activities (http LT4eL). The developed technology will facilitate access to learning objects and will support decentralization and cooperation in content management. To a large extent, this is achieved by extending Learning Management Systems with new, language technology based functionalities, namely a keyword extractor a glossary extractor and a semantic search tool.

At some point in their internal workflow, these new functionalities resort to the outcome of shallow language processing tools ranging from sentence chunkers to lemmatizers and including POS taggers and morphological analyzers. The design constraints impinging on this access set a clear case for the deployment of web services that should be able to deliver the expected outcome of those tools:

- new applications or functionalities build on the outcome of other subordinated applications or tools;
- new versions of the subordinated tools with improved performance should be seamlessly made available with no extra development effort on the side of the client application;
- the licensing of copies of the subordinate tools is not an option available for the stakeholder of the new client application, but the licensing of the outcome of those tools, as delivered by a remote internet-based service, can be.

Though immediately motivated by a specific need to made available language technology for Portuguese for a particular purpose, the work reported in the present paper helps to shed light on what we deem to be an important paradigmatic shift in the way language technology is made available, and most likely in terms of the overall impact of this technology with respect to its neighbouring areas. This is the issue addressed in the next Section 2.

In Section 3, we introduce the processing tools for Portuguese that are being resorted to in order to support the deployment of the web services. The web services and their development are described in Section 4. In Section 5, we discuss the work in progress that is currently being undertaken, and finally in Section 6 we present concluding remarks.

2. Language technology via web services

From the initial design plan up to the final users, the production chain of language resources encompasses several stages, where development tools, validation methodologies, or encoding standards, among many other

instruments, are called to play a role. Aiming at improved production chains, – either in terms of efficiency or in terms of the quality of their outcome –, almost all such stages and supporting instruments have been under extensive development in the recent past. This is attested, for instance, by the burgeoning activity being reported in the LREC conferences, to refer just one of several signs of evidence.

One of the key stages in the language resources production chain is the distribution step. This step involves non negligible issues, ranging from intellectual property to versioning aspects just to refer a few, that in their more structured setups, have found important support by distribution agencies such as the European Language Resources Association (http ELRA) or the Linguistic Data Consortium (http LDC). Also this stage of distribution is receiving closer attention and being developed so that it may get raised to a new level of organization that improves its added value, by exploiting the full potential of web technologies.

Fostered by projects like DAM-LR (http DAM-LR) or initiatives like CLARIN (http CLARIN) and Language Grid (http Language Grid), what is being pursued is the distribution of language resources via web services that grant seamless access to human users and, crucially, also to client applications — by automatically handling their authentication and authorization privileges, tuning to their profile and preferences, managing licensing issues, selecting convenient personalized versions, etc.

While this reshuffling of the distribution of language resources will be taking shape in the near future, the direction of its evolution is inspiring enough to induce similar reshuffling trends in neighbouring areas, in particular in what concerns language technology. In this respect, the granting of access to the language technology tools as web services may be the key to bringing the distribution stage of these tools to a level of dissemination similar to the one achieved for language resources.

In this connection, it is reasonable to anticipate that the availability of language technology tools as web services will have an even deeper impact than a similar move will have on the side of language resources. In tandem with the web service-oriented distribution of resources, this can be the stepping stone for a whole new stage of more widespread incorporation of natural language technology in the semantic web itself. Note that the key issue here is not on taking the language technology as a component of semantic web services, but on the language processing tools being supplied under the form of web services themselves to client applications (which of course will enhance their availability as components for semantic web services in other domains).

In particular, and narrowing our focus of attention to our field, this new form of availability of language technology tools will have a major positive feedback in the production chain of language resources itself. It will enhance more rapid and accurate production of language resources, very likely by several orders of magnitude.

3. Language technology tools

Against the background discussed in the previous Section, and having as immediate motivation the integration of our language technology tools in the LT4eL project, we worked towards making a range of tools for Portuguese available under this new approach. These tools are a subset of the tools that we have been developing for the processing of Portuguese and they were selected because they satisfy a number of features that are likely to make them more suitable for initial experimentation: They are fast, robust, the linguistic information in their output is well understood, and they perform at state of the art accuracy. They include the following individual tools, covering analysis and generation procedures:

Sentence chunker: detects and marks paragraph and sentence boundaries; 99.94% accuracy

Tokenizer: segments text into tokens, expands contractions, detaches clitic pronouns from verbs, etc.; 99.72% accuracy

POS tagger: assigns POS tags to tokens in context; 96.87% accuracy

Nominal featurizer: assigns inflection features (gender and number) to words from the nominal POS categories, resolving ambiguity in context; 91.07% f-score.

Nominal lemmatizer: assigns a lemma to words from the nominal POS categories (viz. common nouns and adjectives), resolving ambiguity in context; 97.67%

Verbal featurizer and lemmatizer: assigns inflection features (tense, person and number) and lemma (infinitive form) to verbs, resolving ambiguity in context; 95.96% f-score.

Verbal conjugator: delivers a conjugation table given any (attested or putative) infinitive verb form and the specification of possible associated clitics

Nominal inflector: delivers an inflected form given another (attested or putative) form and the inflected features required for the output

For a detailed description, analysis and evaluation results of these tools, see (Branco and Silva, 2004, 2006; Silva, 2007; Nunes, 2007).

4. Web services

A web service is a software application that, crucially, supports direct interaction with other software applications over the Internet (Alonso *et al.*, 2005, Ch.5). In its current stage of development, the LXService for language technology of Portuguese is a web service that already

supports the direct interaction of their clients with three of the above listed tools via the Internet, namely: the sentence chunker, the tokenizer and the POS tagger. Basically, given an input text, the clients can interact with these tools via the web service in order to obtain the outcome produced by them, that is a new version of the input text after its being linguistically annotated up to the level that these tools can ensure.

The access to each one of the tools is granted via different methods — `String chunks(Sring text)`, `String tokenizes(String text)`and `String posTags(String text)` — that are members of the Java class `LXClient`. These methods are invoked over an object of this class. Its constructor requires one parameter related to the authentication of the client before the LXService, namely the client's username as this is registered at the LXService database of clients.

This class is part of the package `pt.ul.fc.di.nlx.lxServiceClient`, which includes also the classes for Exceptions and for the class responsible for the authentication before the LXService. Hence, for making an application that is a client of LXService to work, one just needs to get hold of a copy of this package, after the authorization for the utilization of the service being granted and a username and password had been assigned. A summary of the API for the key client class of the LXService, `LXClient`, is displayed in Annex A.

At the service endpoint, in turn, the web services offered by the LXService are implemented in a Java class whose methods are remotely invoked by clients and that locally call the appropriate chain of processing tools on the server side. These services run on a web server supported by Apache Tomcat. To implement such services, the SOAP (Simple Access Object Protocol) was used to transfer the data between them and their clients, in particular with the implementation provided by Apache Axis. Local copies of the Axis libraries at the client endpoint are thus also required, together with the `pt.ul.fc.di.nlx.lxServiceClient` package.

The authentication is ensured by the implementation of Web Security Service, provided by Apache WSS4J, which is endorsed by the OASIS security standard.

The key functionality of a web service — that it supports direct interaction among applications over the Internet, – however, is not in conflict with the fact that further, online services may be offered to human users as well. Accordingly, in order to enhance their visibility and extend their added value, in previous projects we have been progressively rendering them also under a presentation layer that permits their usage directly by human users, e.g. to support second language learning, basic linguistic research, etc. Hence, the tools presented above were bundled into four suites of self-contained functionality as described below, whose front pages are depicted in Annex B:

LX-Inflector,
`http://lxinflector.di.fc.ul.pt`
This service allows its clients to obtain the inflected form from any nominal form provided (common nouns or adjectives — neologisms included), and according to a given specification of inflection feature values entered. This functionality is supported by the composition of the nominal featurizer, lemmatizer and inflector tools indicated above. These tools are implemented in C and Java, and combined through a shell script run by JSP, which interfaces this service with its clients.

LX-Lemmatizer,
`http://lxlemmatizer.di.fc.ul.pt`
This service provides fully-fledged lemmatization of Portuguese verbs: given a verb form (neologisms included), it delivers all possible lemmata together with their respective inflection features. The supporting verbal lemmatization tool was implemented in Java. It is invoked by the clients via JSP.

LX-Conjugator,
`http://lxconjugator.di.fc.ul.pt`
This service provides fully-fledged conjugation tables for any verb form entered (neologism included), including the full range of pronominal forms. The underlying conjugator tool is implemented in Python, which is invoked via CGI.

LX-Suite,
`http://lxsuite.di.fc.ul.pt`
This service is supported by the composition of the range of processing tools for analysis: sentence chunker, tokenizer, POS tagger, nominal featurizer, nominal lemmatizer and verbal lemmatizer. The resulting functionality ensures that the input, entered as raw text, is sentence and token-segmented, their tokens are associated with corresponding lemmata and tagged with linguistic information on their POS and inflection feature values. The outcome is resolved with respect to ambiguity that arise at the different levels of processing. Most of these tools are implemented in C. They are pipelined together through a shell script that is invoked by clients via CGI.

5. Work in progress

The web services just described were put together by resorting to mature web technologies. While offering the expected robustness to the services supported, these technologies fall short of addressing some of the key issues that emerging technologies for web services are aimed at addressing. As of now: the location of these services, as well as their functionality were made known via postings to email lists of communities whose users have interests that are close to language technology — which imposes that client applications still need to be tuned "by hand" if they are to resort to them.

Very interestingly, and encouraging, even under this first setup, and without an API yet that can be resorted to

for the principled invocation of some of these tools, their experimental version as online versions for human has not hampered that these versions have already been accessed by client applications. Apparently, their programmers have found that the added value offered by these services are worth enough to justify going into the work of getting at their outcome via bare scrapping of the HTML document it is embedded into.

At the moment of printing the present paper, the LXService is planned to be expanded so that further methods are included in the client `LXClient.class` which grant access to the remaining processing tools listed above, namely those concerned with lemmatization and morphological analysis. This is a step to be taken with the benefit of the experience and feedback gathered with the running of the current version of the implemented services.

LXService is also being worked out in order to evolve towards further stages that integrate technologies and standards that are emerging specifically to support web services. This includes the utilization of WSDL language to describe these services and allow for their automatic finding (in tandem with future mature UDDI registers).

6. Concluding remarks

Given the experience gained so far, and even only with the current preliminary version, a couple of key issues concerning the viability of the sought for paradigm of language technology tools as web services have emerged. And they are very likely to persist even through the upcoming versions that integrate the above mentioned specific technologies for web services.

As can be checked upon using the LX-Suite online service, our tools are delivering their core outcome in an internally defined format. It will be straightforward to convert from this format into any other format established as a standard (expectedly on top of XML). In spite of its utmost convenience, and of previous and current efforts, no encoding schema for linguistic information associated to natural language expressions has gained momentum as a *de facto* standard.

Another issue concerns authentication and access by trusted clients and copy rights management. As the language technology tools as web services paradigm expands and gains momentum (or even to make this happen), it is likely that providers of high quality services will be willing to join in case they can't hold control over these aspects. Broadly taken, these types of questions are not restricted to the class of web services we are establishing. They appear to be the incarnation in this realm of language technology of similar problems that the deployment of service-oriented computation is facing (Alonso *et al.*, 2005, Ch.9).

Among the different routes that are envisaged as helpful to overcome these problems, there is one that we think may well be suited for our realm of language resources and technology. One of the possible ways out is that a major player, given its dimension and importance, induces the establishment of *de facto* encoding standards (the schemas that that entity eventually adopts). Moreover, if that entity is reliable and independent from service providers and consumers, it will be accepted as a trusted broker for the contracting of services among interested parties and for the certifying of copy rights compliant transactions. The research infra-structure whose creation is a major goal of CLARIN is a most prominent candidate to evolve towards an entity that can play such a role.

7. References

Branco, António and João Silva, 2004, "Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese". Proceedings LREC2004.

Branco, António and João Silva, 2006, "Dedicated Nominal Featurization of Portuguese", *Lecture Notes in Artificial Intelligence*, 3960, Springer.

http CLARIN - http://www.mpi.nl/clarin/

http DAM-LR - http://www.mpi.nl/DAM-LR

http ELRA - http://www.elra.info

http Language Grid - http://langrid.nict.go.jp

http LT4eL - http://www.let.uu.nl

http LDC - http://www.ldc.upenn.edu

Silva, João, 2007, *Shallow Processing of Portuguese: From Sentence Chunking to Nominal Lemmatization*, MSc Dissertation, University of Lisbon.

Nunes, Filipe, 2007, *Verbal Lemmatization and Featurization of Portuguese with Ambiguity Resolution in Contex*t, MSc Dissertation, University of Lisbon.

Annex A - API (summary) of `LXClient`

pt.ul.fc.di.nlx.lxServiceClient

Class LXClient

```
java.lang.Object
  └pt.ul.fc.di.nlx.lxServiceClient.LXClient
```

```
public class LXClient
extends java.lang.Object
```

Client of the LXService, a web service of language technology for Portuguese.

Version:
1.0 (2008-03-07)

Author:
NLX-Natural Language and Speech Group of the University of Lisbon, Department of Informatics

Constructor Summary

`LXClient(java.lang.String username)`	Creates an LXClient object.

Method Summary

`java.lang.String`	`chunks(java.lang.String text)` Segments into sentences and paragraphs with **LX-Chunker**. Marks sentence boundaries with `<s>...</s>` and paragraph boundaries with `<p>...</p>`. Unwraps sentences split over different lines. See: accuracy of LX-Chunker.
`java.lang.String`	`posTags(java.lang.String text)` Segments into sentences and paragraphs with **LX-Chunker** and into lexemes with **LX-Tokenizer**, and annotates with POS tags with **LX-Tagger**. Assigns a single morpho-syntactic tag, from the tagset below, to every token.
`java.lang.String`	`tokenizes(java.lang.String text)` Segments into sentences and paragraphs with **LX-Chunker** and into lexemes with **LX-Tokenizer**. Tokenizes text into lexically relevant tokens.

Annex B - GUIs of the online LX tools